## ****1. 1D and 2D Arrays****

### 🔹 Assignments

1. Write a program to find the **largest and smallest element** in a 1D array.
2. Take an array of n integers as input and **reverse the array** in place.
3. Write a program to **merge two sorted arrays** into a single sorted array.
4. Input a matrix and print its **transpose**.
5. Write a program to **multiply two matrices**.
6. Write a program to calculate the **sum of each row and each column** of a matrix.

### 🔹 Interview Questions

* How are arrays stored in memory in C?
* Difference between **row-major** and **column-major** order.
* What happens if you access an array out of bounds in C?
* How do you pass a 2D array to a function in C?
* Difference between **static arrays** and **dynamic arrays**.

## ****2. String Manipulation and Functions****

### 🔹 Assignments

1. Write a program to **reverse a string without using library functions**.
2. Check if a string is a **palindrome**.
3. Count the number of **vowels, consonants, digits, and spaces** in a string.
4. Write a function to **compare two strings manually**.
5. Remove all **duplicate characters** from a string.
6. Find the **longest word** in a sentence.

### 🔹 Interview Questions

* Difference between **character array** and **string in C**.
* What is the significance of the '\0' character in strings?
* How does scanf("%s", str) behave differently from scanf(" %[^\n]", str)?
* What are common pitfalls of handling strings in C?
* Explain how strings are passed to functions in C.

## ****3. Character Arrays vs char\*****

### 🔹 Assignments

1. Declare a string using **char array** and another using **char\***. Modify both—what happens?
2. Write a program showing **pointer arithmetic** with char\*.
3. Create a function that takes a **char\*** as argument and prints the string length without using strlen.
4. Store "Hello" in char str1[] and in char \*str2. Try modifying both. Document the results.

### 🔹 Interview Questions

* What is the difference between char str[] = "Hello"; and char \*str = "Hello";?
* Why can’t we modify string literals in C?
* How is memory allocated differently for char arrays and char\*?
* Explain the difference between **stack allocation** and **read-only data section** for strings.
* How does const char\* help in safe string handling?

## ****4. String Library Functions****

### 🔹 Assignments

1. Use strlen to find the length of a user-input string.
2. Use strcpy to copy one string into another.
3. Concatenate two strings using strcat.
4. Compare two strings using strcmp.
5. Use strchr and strstr to search for a character and substring.
6. Use strtok to split a sentence into words.

### 🔹 Interview Questions

* What is the difference between strcmp, strncmp, and strcasecmp?
* Why is strcpy unsafe? What should be used instead?
* How does strtok work internally? Why is it not thread-safe?
* Difference between memcpy and strcpy.
* What is the difference between strcat and strncat?

# 🎯 Suggested Practice Flow

1. Start with **1D and 2D arrays** (basic data handling).
2. Move to **string manipulation** (manual implementations).
3. Study **char arrays vs char\*** (memory-level understanding).
4. Practice **string library functions** (standard functions + pitfalls).